I Digital Signal Processors 1

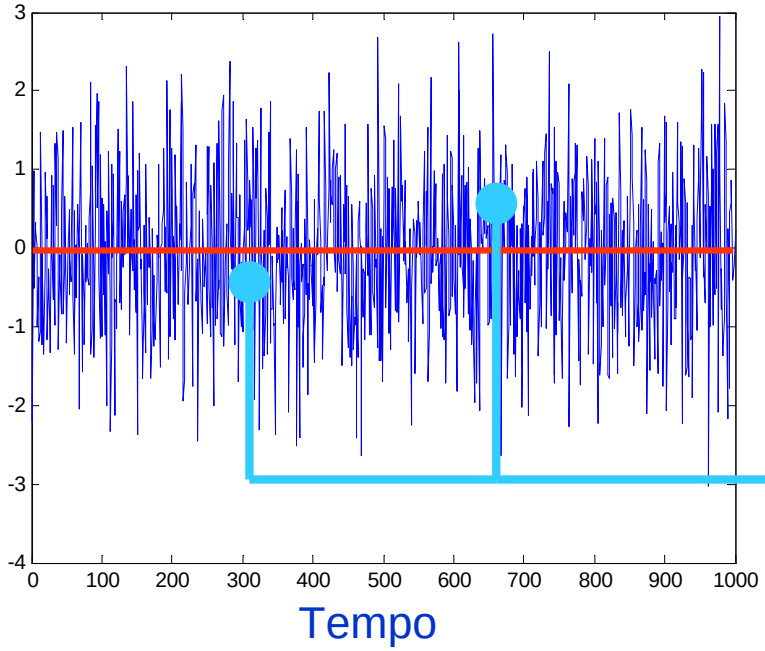
martedì 11 aprile 2006

I DSP in termini rozzi possiamo definirli come dei microcontrollori ottimizzati per avere prestazioni spinte nella realizzazione di filtri di segnali analogici, ovviamente convertiti digitalmente.

Dal punto punto di vista strutturale sono organizzati in modo da sfruttare in pieno l'organizzazione della memoria di tipo Harvard.

Per capire le ragioni delle scelte strutturali dei DSP occorre avere un'idea del tipo di applicazione tipica che si richiede a questi dispositivi.

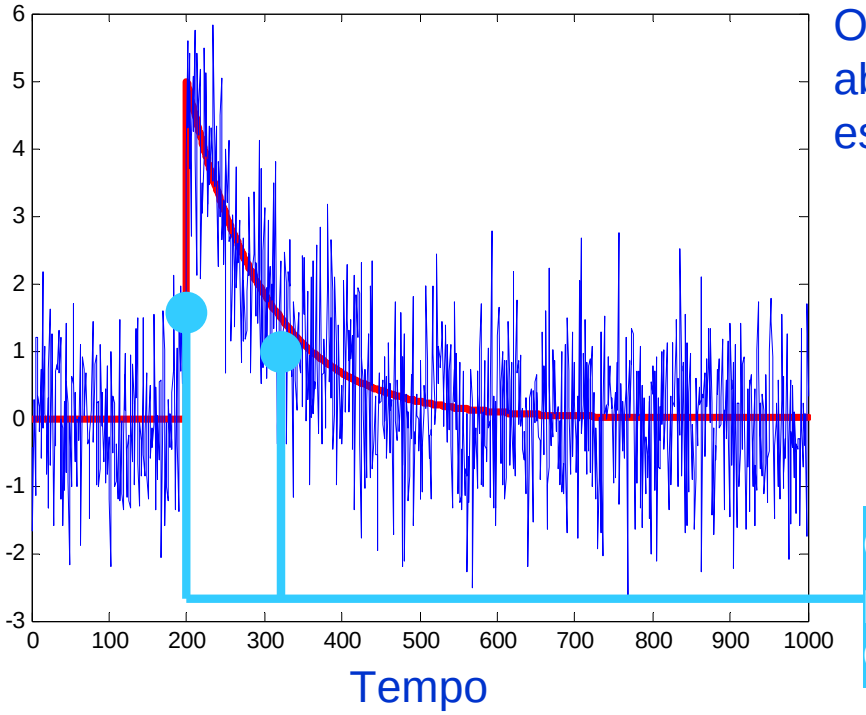
Supponiamo di dovere trattare un segnale analogico che abbia sovrapposto dei disturbi che possono infastidire la sua interpretazione. Un esempio tipico è fornito da un segnale vocale che proviene da una stazione radio con sovrapposto i tipici 'gracchi' di una non perfetta sintonizzazione.



Qui a fianco abbiamo un tipico esempio di una linea che presenta solo rumore. Ci aspetteremmo di leggere un segnale nullo ad ogni istante. In realtà abbiamo un segnale casuale di disturbo.

E' difficile pensare che vi sia una correlazione tra i 2 valori di rumore misurati agli istanti indicati.

Il compito principe di un DSP 1



Ora al rumore precedente abbiamo sovrapposto un segnale esponenziale di interesse.

Ora notiamo che tra i 2 istanti presi in considerazione esiste un certa correlazione.

Un modo per cercare di attenuare gli effetti del rumore e risaltare il segnale di interesse di forma nota, ma ampiezza incognita, è quello di pesare i campioni acquisiti nel tempo con quelli che appartengono al passato: la parte del segnale che si trova correlato avrà un contributo maggiore del rumore che si presenta casualmente.

Sia: $V(-n\Delta t)$, $V(-(n-1)\Delta t)$, $V(-(n-2)\Delta t)$, ….$V(0)$, il migliore modo per ottimizzare l'interpretazione del segnale è trovare un insieme di coefficienti b_n, b_{n-1}, b_{n-2}, … b_0 capaci di mettere in evidenza la correlazione tra i campioni letti secondo la legge:

$$V_{Fil}(0) = \sum_{K=0}^{n} b_K V(-K\Delta t)$$

Ad ogni lettura il valore più vecchio viene perso, tutti gli altri valori letti 'invecchiano' di una posizione ed il valore letto si chiamerà v(0).

La relazione trovata è invero molto più generica. Vale anche nel caso in cui il segnale non abbia forma nota ma sia la sovrapposizione di più onde. Cambierebbe solo il criterio di calcolo dei coefficienti, che viene fatto una volta sola, prima dell'inizio del processo di filtraggio, nella fase di stesura del firmware.

Il compito principe di un DSP 2

Quindi il processo consiste in:

1. Campionamento del segnale analogico ad intervalli regolari;
2. Conversione dei campioni letti in valori digitalizzati;
3. Applicazione del filtro consistente in una somma di prodotti con coefficienti noti;
4. riconversione di ogni campione trattato in un nuovo segnale analogico.

L'aspetto fondamentale è pertanto quello di implementare una funzione consistente in una somma di prodotti:

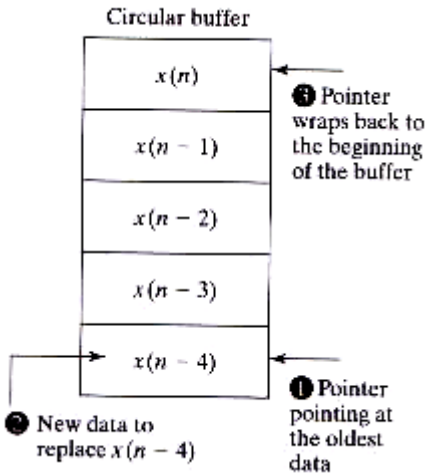
$$V_{Fil}(0) = \sum_{K=0}^{n} b_K V(-K\Delta t)$$

In cui i coefficienti b_k sono noti, mentre i valori $V(-K\Delta t)$ sono letti dal mondo analogico e convertiti in numeri.

La proprietà fondamentale di un DSP è proprio quella di compiere con grande efficienza e velocità le operazioni di moltiplicazione e somma.

Di fatto la struttura HD di un DSP ricalca proprio questa proprietà.

Il buffer circolare ed il bit reversal



La continua acquisizione dei campioni richiede un aggiornamento del vettore che contiene la 'storia' del segnale. Per utilizzare un numero costante di celle di memoria, viene usato un contatore apposito che si aggiorna in modo circolare in modo tale da buttare via il campione più vecchio sostituendovi quello più recente.

Praticamente tutti i DSP possiedono una funzione apposita che si occupa di implementare questa procedura.

Nel computo della FFT (Fast Fourier Transform) si usa un algoritmo che, sfruttando un ordinamento razionale dei campioni, consente di risparmiare il numero di moltiplicazioni da effettuare.

Il risultato della procedura è un vettore non ordinato in modo crescente, ma secondo un algoritmo noto. Mediante l'operazione così detta di 'bit reversing' anche questa implementata in quasi tutti i DSP, si riesce a ricostruire l'ordinamento corretto del vettore risultante.

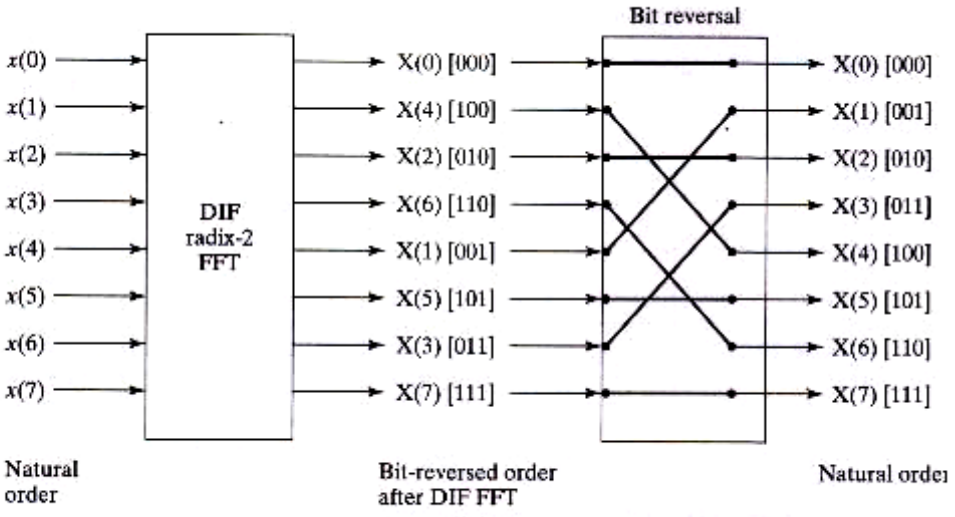


Figure 3.12 Block diagram of the DIF radix-2 FFT (numbers in square brackets are unsigned binary integers)

Esempio di implementazione di un filtro

Nelle operazioni di filtraggio sono importanti le operazioni di MAC, Multiply and Add operation, e MACD, MAC + movimentazione finale dei dati.

The output of an FIR filter of length L is computed as

$$y(n) = \sum_{i=0}^{L-1} b_i x(n-i). \qquad (4.2.2)$$

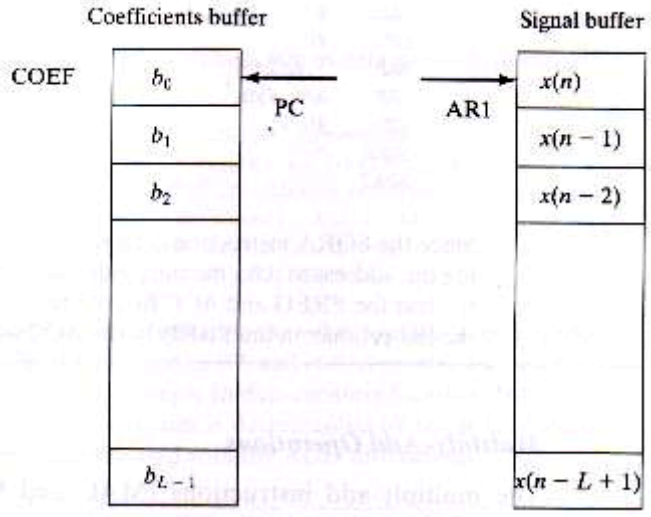
The signal buffer $\{x(n), x(n-1), \ldots, x(n-L+1)\}$ is located in data memory space and is arranged as shown in Fig. 4.2. Assume that COEF is the symbolic address of the coefficient buffer located in the program memory and that the current AR1 points to the beginning of the signal buffer as shown in Fig. 4.7. We can use the following instructions to implement FIR filtering:

```
LACC   #0       ; clear ACC
RPT    #L-1     ; repeat the next instruction L times
MAC    COEF,*+  ; multiply and accumulate previous product
APAC            ; accumulate the last product
```

The MAC instruction multiplies the signal $x(n-i)$ in data memory pointed at by AR1 with the coefficient b_i in program memory pointed at by PC, which is initialized to be pointing at the first coefficient at program-memory address COEF.

Un'altra tipica proprietà del SW DSP è l'istruzione di repeat. In questo modo un filtro come quello sopra viene eseguito in sole 4 istruzioni.

il filtro agisce su 2 vettori: quello dei coefficienti e quello dei campioni letti:



Efficienza del DSP: Pipeline

Si supponga di dividere un'istruzione in più fasi distinte:

1. PF: pre-fetch acquisizione dell'indirizzo dell'istruzione di cui si intende fare il fetch;
2. F: fetch, caricamento, dell'istruzione;
3. D: decodifica dell'istruzione;
4. A: lettura eventuale dell'indirizzo dell'operando;
5. R: lettura del dato;
6. X: esecuzione dell'operazione e scrittura del dato.

A questo punto si eseguono le varie fasi in parallelo in modo che non si sovrappongano e disturbino. Si ottiene così un certo risparmio di tempo. Il livello di pipeline dipende dalla sofisticazione del DSp in considerazione. Nell'esempio indicato abbiamo 6 livelli di pipeline:

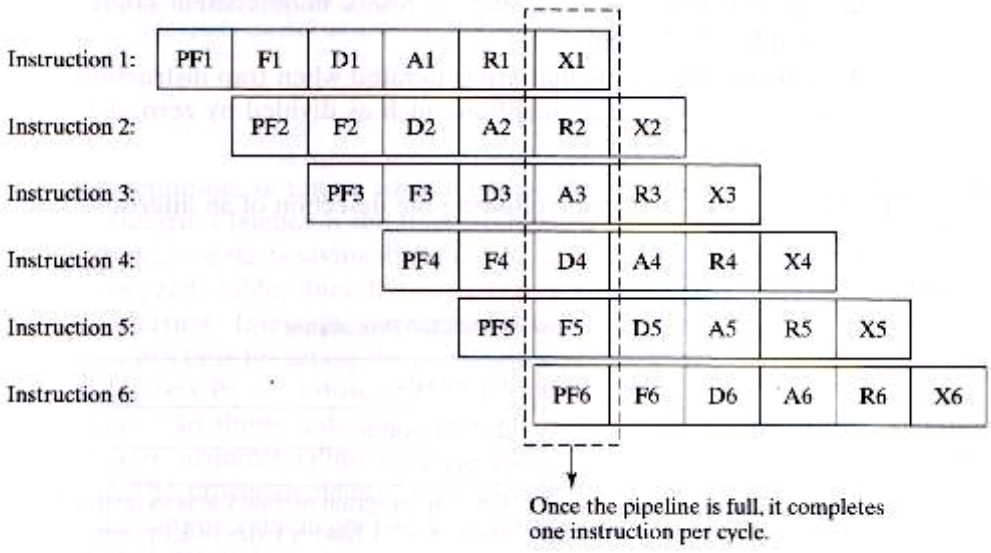


Figure 3.13 Six pipeline stages of the TMS320C54x

Efficienza del DSP: cache

3.4.3 Instruction Cache

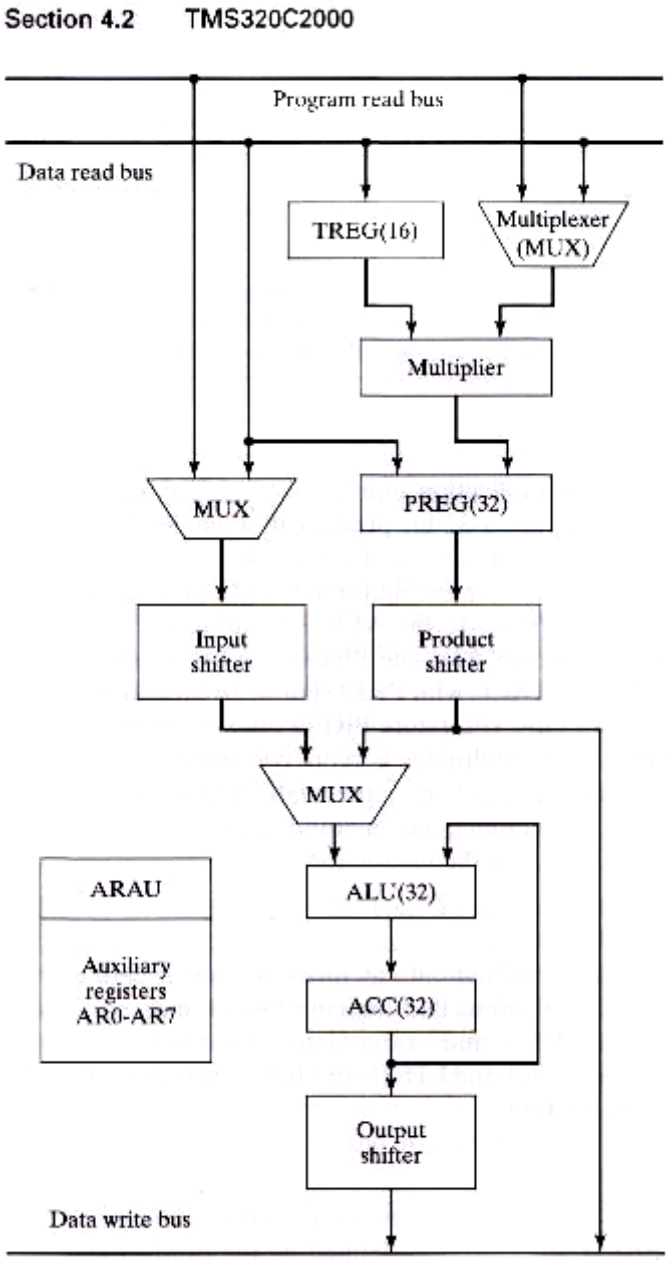
Instruction cache is becoming a popular feature for modern DSP processors. It is a small and fast on-chip memory that is used to store frequently used code, such as the instructions inside a repeated loop. The instruction cache eliminates the time spent in fetching the instructions from the slow off-chip program memory, thus freeing memory access for a data read (or write). Table 3.12 summarizes the availability of cache memory and their sizes for different TMS320 processors.

Qui abbiamo 2 aspetti importanti da prendere in considerazione. Un DSP deve essere sempre il più veloce possibile onde fornire risultati in tempo reale.

Per tale ragione i DSP non presentano quasi mai a bordo una memoria FLASH o ROM per i programmi perché in genere troppo lente. Si affidano a memorie esterne da cui si copia il programma in una RAM interna veloce alla partenza del sistema.

La memoria cache non serve solo a contenere passi di programma che devono essere ripetuti in sequenza. Spesso la si sfrutta per non impegnare il bus dati ed indirizzi che può così essere sfruttato pienamente per il trasferimento degli operandi.

L'unità centrale



I DSP sfruttano una modifica della struttura Harvard. Ad esempio Analog chiama la serie dei propri DSP SHARC, che sta per Super Harvard Architecture.

Per esempio nei DSP Texas si usano 6 bus:

1. Program-address bus;
2. Data-read-address bus;
3. Data-write-address bus;
4. Program-read bus;
5. Data-read bus;
6. Data-write bus.

Ovviamente la memoria dati è separata da quella dei programmi. Inoltre si può osservare che un operando si può leggere dal program-read bus. Questo perché i coefficienti dei filtri li si fa risiedere nella memoria programmi.

In tutti i DSP è implementato un blocco di moltiplicazione che precede quello di somma. Così in HD con un solo colpo di clock si riesce a svolgere un'operazione di moltiplicazione in contemporanea ad una somma.

Esiste un blocco di registri chiamati ARAU che servono in special modo nell'indirizzamento indiretto nella realizzazione dei filtri.

Un DSP completo Fixed-point: Texas TMS320C54x

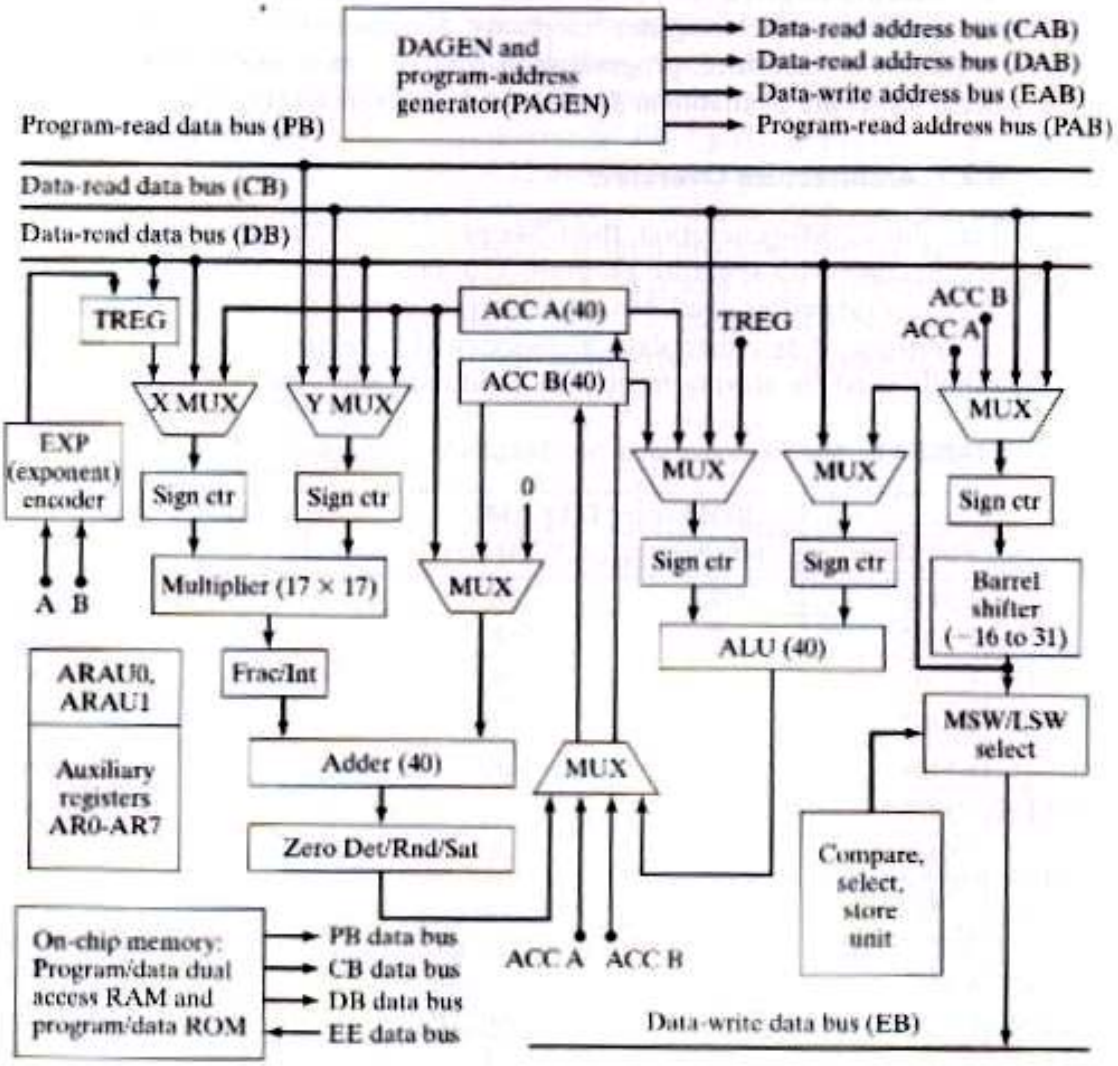


Figure 4.13 Block diagram of the TMS320C54x internal hardware

Si riconosce la struttura Harvard modificata dalla presenza di molti bus per il trasferimento dei dati e delle istruzioni.

Gli indirizzi sono generati dal PAGEN per quanto riguarda le istruzioni ed il DAGEN per quanto riguarda i dati.

Il barrel shifter serve a scalare i dati, mediante shifting in un singolo colpo di clock, onde evitare la saturazione come risultato delle somme e delle moltiplicazioni. Ogni shift a destra comprende la divisione per 2 del dato. Ovviamente la divisione del dato implica una perdita di informazione.

Il moltiplicatore a singolo colpo di clock è ovviamente presente.

Un DSP completo Fixed-point: Texas TMS320C5510

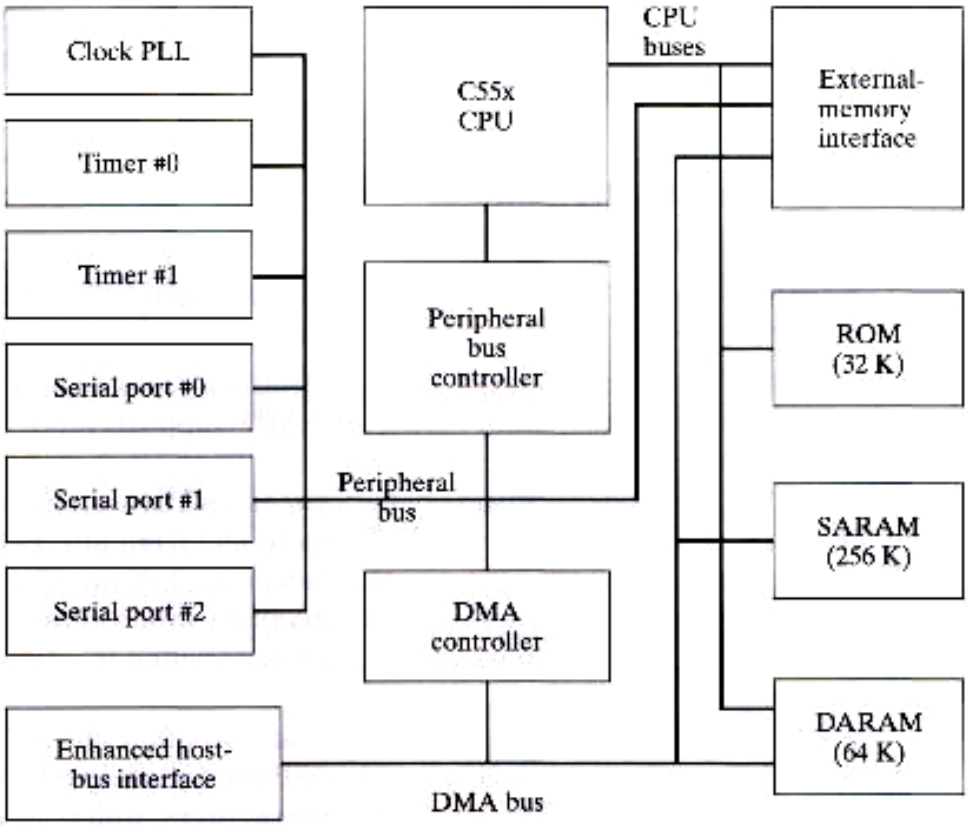


Figure 4.21 Block diagram of the TMS320C5510

Lo schema è più generale qui.

Di interessante abbiamo la SARAM che sta per single access RAM. Ovvero ad ogni colpo di clock in questa memoria si può leggere o scrivere. Ma c'è anche la DARAM che sta per double access RAM. Ovvero ad ogni colpo d clock può essere doppiamente indirizzata per una contemporanea lettura e scrittura di dati.

Una cosa molto importante nei DSP è la presenza della modalità di indirizzamento così detta DMA, Direct Memory Access. Secondo questa modalità la CPU non interviene nell'indirizzamento. Questo significa che una periferica può scrivere direttamente nella memoria adibita in DMA senza aspettare il servizio della CPU. Questo è molto utile quando i dati di ingresso provengono da un campionatore. Mentre la CPU elabora il filtraggio l'ADC può scrivere direttamente il dato campionato nella memoria.

I floating-point DSP: Texas TMS320C30

I DSP floating-point consentono l'elaborazione dei dati anche in fixed –point. Questo ovviamente perché il segnale proveniente dall'ADC o trasmesso al DAC deve essere fixed-point.

I DSP floating-point consentono pertanto la possibilità di selezione della modalità di trattazione del dato, ovvero possono convertire dati da fixed-point a floating-point e viceversa.

La modalità di trattazione è completamente trasparente al progettista, tanto che la distinzione tra DSP floating-point o fixed-point si riesce a verificare solo dalle specifiche del dispositivo o dal fatto che nel set di istruzioni è presente l'istruzione per la conversione dei dati.

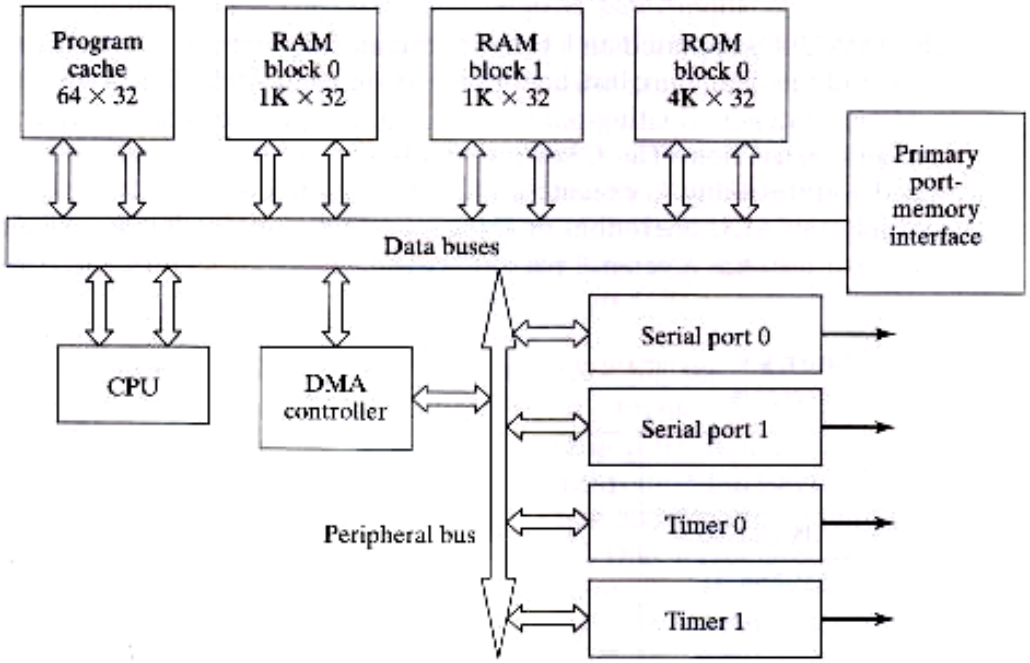
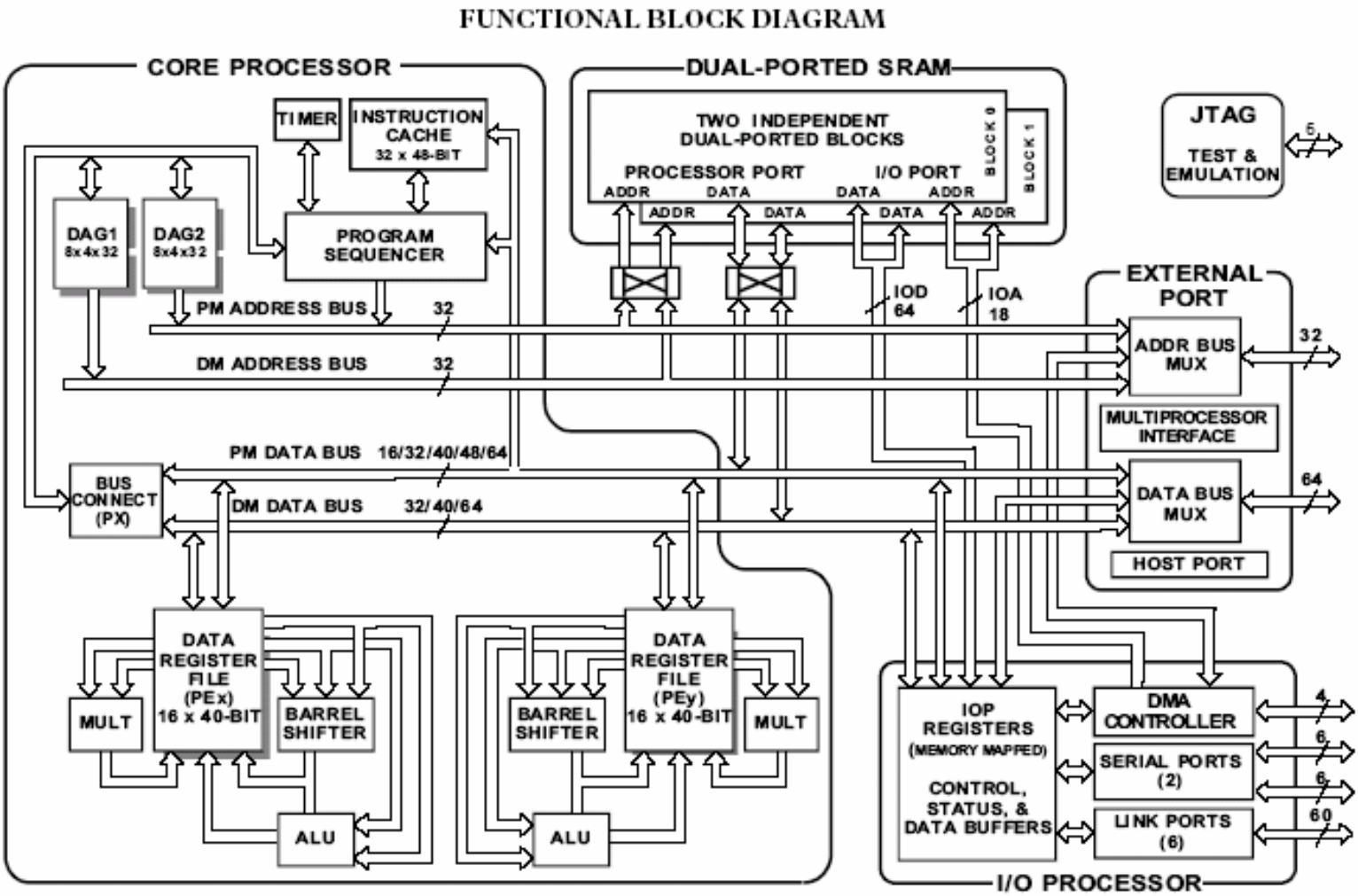


Figure 5.1 Block diagram of the TMS320C30

Per esempio nello schema a blocchi di questo DSP floating-point di Texas non si scopre la presenza della modalità floating-point.

In genere i DSP posseggono più di una seriale capace di trasmettere molto velocemente. Infatti la connessine con il convertitore audio, detto CODEC, per esempio, deve avvenire con l'ADC ed il DAC. La doppia seriale consente la trasmissione contemporanea dei dati in ingresso ed uscita, in modalità DMA.

I floating-point DSP: Analog Devices SHARC ADSP-21160M 1



Questo DSP è stato progettato per trattare segnali audio. Ha una doppia ALU per consentire l'elaborazione dei 2 canali stereo contemporaneamente.

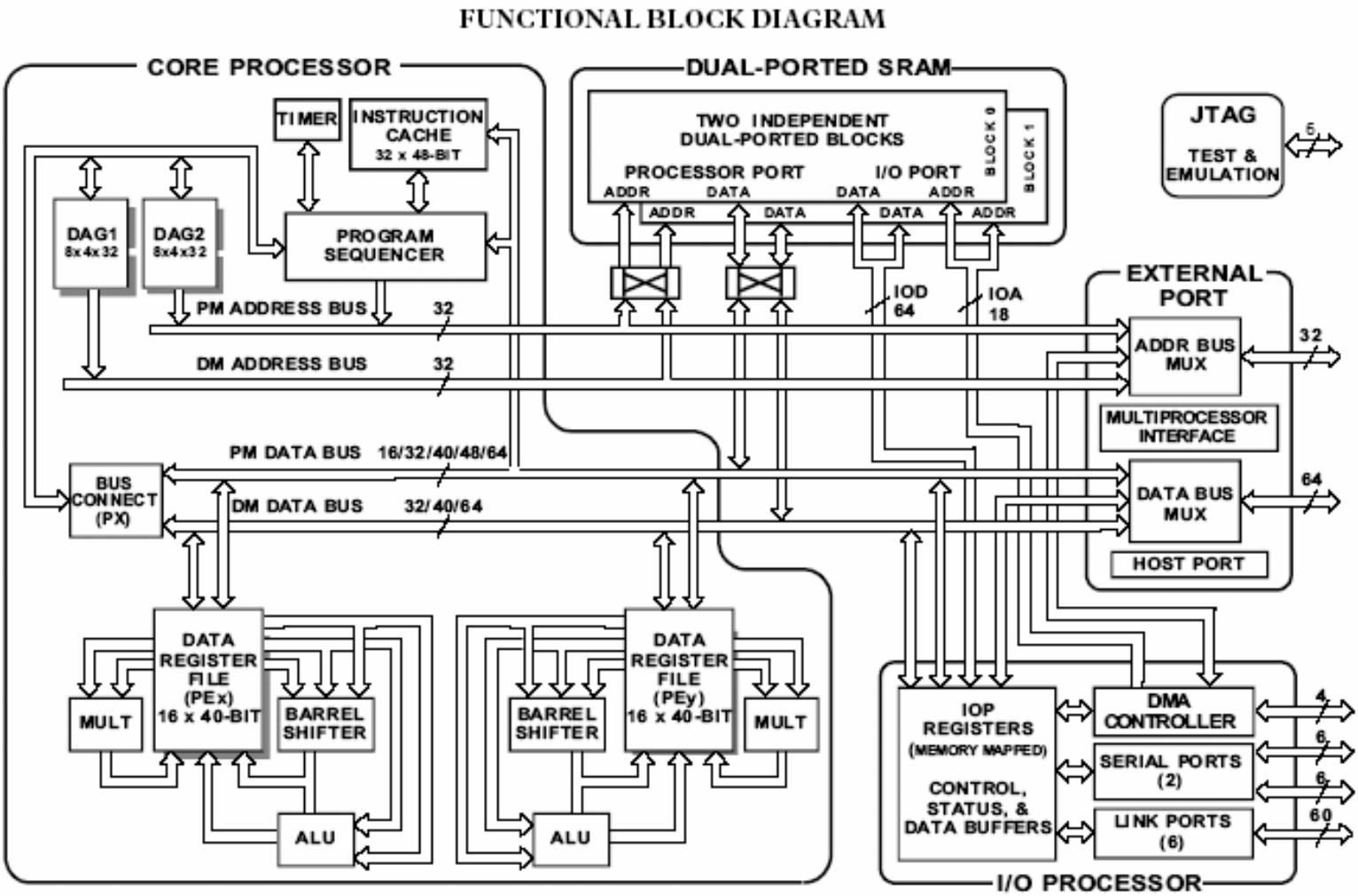
La struttura è Harvard modificata con 4 bus: il bus di indirizzo per le istruzioni, PM address bus, il bus di indirizzo per i dati, DM address bus, quindi ha anche il bus dati, DM data bus, ed il bus istruzioni, PM data bus. Occorre però considerare che il PM data bus viene anche usato per leggere il secondo operando, il coefficiente del filtro.

Gli indirizzi vengono generati indipendentemente in 2 DAG, Data Address Generator.

Possiede una memoria veloce, SRAM, da 4 Mb divisa in 2 banchi da 2 Mb. In un banco viene caricato il programma dalla FLASH esterna allo start-up,che può così essere manipolato velocemente.

La SRAM dati può essere accessibile mediante DMA anche dalle periferiche esterne.

I floating-point DSP: Analog Devices SHARC ADSP-21160M 2



Per potere usare il PM bus come bus dati è presente una cache che consente di mantenere fino a 48 istruzioni mentre il PM bus resta impegnato per il trasferimento di dati.

La programmazione e l'emulazione del DSP può essere condotta mediante interfaccia JTAG, che è un sistema di comunicazione che consente la verifica così detta in-circuit, ovvero con il dispositivo nella configurazione circuitale finale.

Le unità aritmetiche sono corredate del Data Register File che consente di memorizzare dati e risultati delle operazioni, una specie di cache per i dati.

Va osservato che avendo 2 unità di calcolo operanti in parallelo il trasferimento dei dati è organizzato in modo tale da potere trasferire 4 dati per singolo colpo di clock.

Bibliografia

Sen M. Kuo, Woon-Seng Gan

Digital Signal Processors, Architecture, Implementations and applications

Pearson Prentice Hall. Cod. Biblio: 621.3822 KUOS.DIG/2005

Datasheet Texas Instruments.

Datasheets Analog Devices.